TWO LEVEL BOOLEAN MINIMIZATION USING A SMALL DIGITAL COMPUTER

By

ROBERT WHITNEY BUTLER

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1959

Submitted to the faculty of the Graduate College of the Oklahoma State University in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE May 1966

TWO LEVEL BOOLEAN MINIMIZATION USING A SMALL DIGITAL COMPUTER

Thesis Approved:

Paul A. McCollum
Thesis Adviser

Kenneth A. McCollom

J. K. Bragg
Dean of the Graduate College

ACKNOWLEDGEMENTS

I left the Oklahoma State University campus with the intention of finishing my thesis while working for International Business Machines Corporation. I wish to thank my managers at IBM whose promptings made it necessary to live up to my intentions. My hope is that any other student who commits the same folly has managers as understanding as mine.

I am very grateful to Professor Paul A. McCollum for his extra effort beyond the call of duty taking care of my problems on campus as I was not there to attend to them myself and for his regular capacity as my advisor.

I want to acknowledge the support of IBM in extending me the use of their 1620 and reproduction facilities and allowing me time away from my work to complete parts of my thesis.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

The development of the electronic computer has called for the use of larger and more complicated logic expressions. These expressions have called for some means of handling them by automatic methods to eliminate the human drudgery and error in reducing the expression as given to an expression that represents a minimum cost circuit.

George Boole introduced the algebra of logic, now called Boolean Algebra, in 1847 in a mathematical paper dealing with the analysis of logic. From this algebra has evolved classical techniques of reducing an expression to a more minimal form. Some of the well known techniques are the Venn diagrams (1), the Veitch diagrams (2), the Harvard chart method (3), and the Quine method (4). The Venn diagram is a topological method of solving the minimization problem. The Veitch diagram is a variation of the Venn diagram in which the method for representing the logic expression has been defined in a systematic fashion so groups of expressions are easily recognized. The Harvard chart and Quine method represent an operational method of handling the problem of minimizing. Since these methods are rather mechanical and repetitious, in their application it is attractive to attempt to amend some of these methods for operation on a computer.

Techniques are being developed on large computers to minimize large logic expressions. This makes available to a group of designing engineers with access to a large computer a means of minimizing their work. There are, however, the engineers which have only a small computer available for use.

This paper deals with the problem of minimizing the boolean expressions on a small computer. The author of this paper was particularly intrigued with the idea of using an IBM 1620. Its addition is done with the use of a table of numbers stored directly in memory, and these numbers can be changed by programming to give results other than straight arithmetical results. The reader will find in Chapter III the outline of techniques used to minimize on a computer. Chapter IV will show the particular method by which the tables were modified in the IBM 1620 to give logical results. Appendix A will give the operating instruction for using the program on the IBM 1620 to minimize a boolean expression.

CHAPTER II

DEFINITIONS OF TERMS

The terminology used in this paper has been developed by Paul Roth (5) and has evolved by usage within IBM. This usage has been presented by T. A. Kircher (6) in a class on logic design to design engineers within IBM.

LITERAL A literal is a variable which has two states. Since a literal often represents the output of one logic element of a computer circuit, the two states in this paper will be the "ON" state represented by a "1" and the "OFF" state represented by a "0".

CUBE A cube is the "AND" expression of its literals. The cube is in the "ON" state if the value of each literal is the state expressed in the cube. The literals are represented in the cube by position. The state of the literal is represented by a "1" or "0". If the literal does not enter into the condition of the cube its position will be represented by a "-" which will be the unspecified condition. An example of a five variable cube would be 10--1. This states that the cube is in the "ON" state if the first literal is "ON" and the second literal is "OFF" and the fifth lit-

eral is "ON". Notice that it does not matter what the state of the third and fourth literals are since they are unspecified.

CUBICAL ARRAY A cubical array is a list of cubes. The cubical array is said to be in the "ON" state if any one of the cubes in the array is in the "ON" state.

VERTEX For a given number of literals "n" in a function there are 2^n possible conditions of the function. A vertex is an expression of just one of these conditions. It is, therefore, written as a cube in which there are no unspecified literals.

COCYCLES If one were to map a cubical array onto a map, such as a Veitch Diagram, a certain area would be covered. If this area were then represented by a list of cubes such that each cube covered as large an area as possible without falling outside the bounds of the original area, each of these cubes would be known as a cocycle. A term which is commonly used for cocycles in classical techniques is Prime Implicants.

SUBSUMED ARRAY A subsumed array is a cubical array in which no cube is completely contained within another cube.

COVER A cover is a cubical array which includes all the required vertices. The cover may also include

vertices which are of no consequence known as dont care terms.

IRREDUNDANT COVER An irredundant cover is a cover in which no cube in the cubical array is completely covered by the other cubes in the array. This is to be distinguished from a subsumed array in which a cube may be covered by the other cubes in the array, but not by any one cube in the array.

SUBSUMING Subsuming is the operation of comparing each cube in a cubical array against all the other cubes in the array to see that it is not contained in any of the cubes of the array. If any cube is covered by another cube it is removed from the cubical array.

SHARPING Sharping is the operation of removing any common area between two cubical arrays from the array on which the sharping is being done. The other cubical array is the array being sharped away. It is interesting to note how this operation happens to carry this particular name. The operation is a subtraction type operation. Paul Roth needed a symbol to denote this operation. The numbers symbol on the typewriter was unused, so he called the operation sharping and used the symbol "#" to denote the operation in his text.

TWO LEVEL MINIMIZATION A cubical array represents a logical circuit of "AND" gates feeding "OR" gates. A mini-

mized circuit would be one which had the fewest number of "AND" gates feeding the "OR" gate, and the fewest number of lines feeding the "AND" gates. In terms of the cubical array this would call for the fewest number of cubes in the cubical array and the largest number of unspecified literals in each cube.

ON TERMS The on terms represent the vertices for which a function must be on.

OFF TERMS The off terms represent the vertices for which a function must be off.

DONT CARE TERMS The dont care terms represent the vertices for which it is unnecessary to specify their state as they will never exist for this particular function.

CHAPTER III

MINIMIZATION TECHNIQUES

For this paper only the two level minimization will be considered for which the criterion will be the minimum cost two level circuit of "AND" gates feeding into "OR" gates. There are two steps to a two level minimization. The first step is to develop all of the cocycles of the given cubical array. The second step is to pick a subset of the cocycles which will give an irredundant cover.

The solution to the first step has been presented in the classical techniques. There are mapping techniques such as Venn diagrams (1), Veitch Diagrams (2), and Karnough maps (2). These mapping techniques give the cocycles by visual inspection after the cubical array has been mapped onto the maps. There is the Harvard chart method which gives a systematic way of cancelling terms out of a chart until the only terms remaining are the cocycles (3). The Quine method is another systematic method which is an operational technique for finding the cocycles (4).

The first three techniques above are topological map methods of handling boolean functions. It is difficult to represent a topological picture in the memory of a computer; however, the greatest problem arises in trying to represent

each vertex in the map. For a large map more memory than is available in a large computer may be needed. For this reason these techniques are not pursued for computer computation.

The Harvard chart is more amenable to computer techniques, but it still suffers from an all possible combinations situation. The Quine method is quite acceptable because of its operational characteristics. However, this method demands that each cube in the cubical array be expanded until all the cubes are vertex cubes. This again involves all possible combinations.

Frank McFarlin (7) presented a variation of the Quine method which did not require that the cubes be expanded to vertex cubes. William Vipraio (8) programmed this method on the IBM 650 for his Masters Thesis at Oklahoma State University.

Paul Roth and associates have presented another method for finding the cocycles (5,6). It is called the double sharp method. It uses as a starting point the universal cube. The universal cube is a cube in which all of the literals are unspecified meaning that it represents all the possible states of the literals. If this cube were represented on a Karnough map it would cover the entire map. The cubical array is sharped from the universal cube. The remaining cubical array is the cocycles of the complement array of the original array. This remaining cubical array is then sharped from the universal cube giving the cocycles of the complement of the complement of the original cubical array

which are the cocycles of the original cubical array. It is this final method which was used in the program written on the IBM 1620 computer. It has several features that make it more attractive to use than the method presented by Frank McFarlin. First, there is less decision making to do in the process of sharping which allows the program to occupy less memory in the computer. Second, the sharping process is used in the second step of the minimization. Since it is required that the program for sharping be written, it is a definite saving to use it for the first step of the minimization.

The second step of the two level minimization is selecting a subset of the cocycles to produce an irredundant cover of the on terms. S. R. Petrick has shown a method suitable for use on a computer for selecting all sets of irredundant covers from the cocycle set (9). The process for selecting the minimum set is left to the individual. In this paper the minimum set has been roughly defined as the lowest cost circuit, which is related to the number of "AND" gates feeding the "OR" gate and, second, the number of legs feeding the "AND" gates. The exact relation of these two costs would depend on a specific circuit design for a two level "AND-OR" gate combination. With this criterion the irredundant covers can be evaluated and the minimum cost cover selected. There is a problem in that the number of irredundant covers can be a very large number.

Paul Roth (5) has presented several methods for selecting a minimum set two of which will follow. His

solutions attempt to cause the computer to converge on a single irredundant cover without generating all irredundant covers. His simplest method is called the Forced Irredundant Cover. To follow the procedure, the process will be traced from the original cubes through the cocycles to the irredundant cover.

First, a cubical array is given which represents the on terms of the function. Second, a cubical array is given which represents the dont care terms of the function. The second cubical array may be empty indicating that the minimized solution may cover only the on terms. If the dont care terms are given, it allows the cocycles to cover the on terms and the dont care terms which allows the cocycles to be larger. The larger cubes will have more unspecified literals and will be more minimal cubes. The cocycles for the on terms and dont care terms are found. After the cocycles are found the selection process will proceed as follows. The cubes, which are the cocycles, will be sorted so the cubes with the least number of unspecified literals are at the head of the cubical array and the cubes with the most number of unspecified literals are at the end of the cubical array. A check is made on each cube to see if it can be eliminated as follows. First, the dont care terms are sharped away from the cocycle under consideration. If the cocycle vanishes at this point, it covers only the dont care terms, and there is no reason to attempt to cover these conditions; therefore, the cocycle is eliminated from the cocycle list. If there

is still area left in the cocycle, all the cocycles which have already been tested and all the cocycles not yet tested are sharped from the remaining area of the cocycle under test. If the area has vanished at this point this cocycle is not needed as the area will be covered by other cocycles. The cocycle is then eliminated from the cocycle list. If there is still area left in the cocycle at this point the cocycle covers part of the on term area that is not covered by any other cocycle and is required in the minimized solution. After all cocycles have been tested, the cocycles which have not been eliminated will be an irredundant cover which was forced by eliminating any cocycle that was found to be redundant at the time that it was tested. This method discriminates against the cocycles which are considered first, since the redundant area is reduced as cocycles are eliminated. However, the cocycle list was purposely sorted so the most expensive cubes would be tested first. This method has the disadvantage that a near minimum solution is not guaranteed. However, experience with this method has shown that the solutions are reasonable. This method does have the advantage that the computation is direct and the solution is obtained very quickly.

Another method is known as the Vertex Count Irredundant Cover. It is an attempt to sacrifice computer time for the sake of being more selective about the cocycles which are retained in the solution. First, the cocycles must be found. The procedure is the same process used in the Forced

Irredundant Cover method. The cocycle list is checked for irredundant cubes. This procedure is again the same process that is used in the Forced Irredundant Cover method except that no cocycle is removed if it is found to be redundant. After the irredundant cocycles have been found, the selection process then begins for picking the least expensive cocycles for the rest of the cover. The cocycle which covers the largest number of vertices of the on terms is chosen for test. A check is made to see that this cocycle is not redundant with the cocycles already chosen. This is accomplished by sharping away all the dont care terms and all the cocycles which are in the solution at this point. If the cocycle vanishes, then it is redundant and is discarded. If not, then it is added to the solution. This process is continued until the original on terms are covered.

From the above methods for finding the minimized cover, the Forced Irredundant Cover method was selected for the 1620 program. This method was used to keep the program as simple as possible and to keep the running time of the program as low as possible.

CHAPTER IV

ALGORITHMS USED ON THE 1620

This chapter is concerned with the actual programming methods of accomplishing specific tasks which make up the program. Major tasks which are performed in different parts of the program are written as subroutines which are then called upon from the main program at the point where these specific tasks are needed. The most important subroutine which becomes the very center of the program by its importance is the sharping subroutine. A second subroutine which is very important is the subsuming routine. The third is the input subroutine, and the fourth is the output subroutine. With these subroutines the main program for finding the two level minimization becomes a straight forward program which links the subroutines together to perform the entire minimization. A reference to Chapter V on the program flow charts will show the reader how these subroutines are tied together. The input and output subroutines being for the purpose of moving information in and out of the computer will also be left to Chapter V for explanation. The sharping subroutine and the subsuming subroutine are the heart of the program and will be explained in detail in the remainder of this chapter.

Before proceeding with the writing of this program it was necessary to pick the computer for which the program was to be written. In looking over the small computers which were available for use the IBM 1620 proved to have an internal feature which made it very attractive to use for this program. It has an add table in addressable storage. A standard way to build a computer is to build an electronic adder which forms the sum of two digits in the computer and puts the answer into some designated area. The 1620, however, approaches the add problem differently. The addition of two digits is a table look-up operation. The two digits to be added together form the last two digits of a position in storage. The computer then goes to this position to get the digit which is the result of the addition of these two digits. If the sum of these two digits causes a carry it is indicated by a flag bit and causes the computer to make an adjustment in adding the next two digits together. Reference to an IBM manual on 1620 operation codes will give this operation in detail.(10) For the answer to be the decimal sum of two decimal digits an appropriate set of numbers must be loaded into the table area which the computer uses. These numbers are supplied in the loading of the program. One may, however, overlay this table with any other set of numbers, causing the addition of two decimal digits to be some number other than the decimal sum.

For internal processing the cubes are represented in a coded form. The "0" is represented by a "1". The "1" is

represented by a "3". And, the "-" by a "5". To achieve the sharping function the add tables have been overlayed with the table shown in Figure 4-1. The blank positions not shown in this table are filled with an invalid number. They do not enter into any part of the sharping operation.

	0	1	2	3	4	5	6	7	8	9
0	0	$\overline{0}$					0	7	0	9
1	1	0	$\overline{0}$	$\overline{0}$	$\overline{0}$	0	$\overline{0}$			
3	3	$\overline{0}$	$\overline{0}$	0	$\overline{0}$	0	$\overline{0}$			
5	5	8	$\overline{0}$	6	$\overline{0}$	0	$\overline{0}$	1		3
6	6							0		
7		$\overline{0}$								
8	8									0
9		$\overline{0}$								

FIGURE 4-1

To illustrate the operation of the special add table cube B will be sharped from cube A. The result will be cube C and cube D. Cube A is represented internally in the

cube A = 01-- cube C = 01-1

cube B = -110 cube D = 010-

machine as 1355 and cube B is represented as 5331. The sharp operation would proceed as follows.

First a transform test is made to see if cube B intersects with cube A and will therefore, remove part of cube A. In the table look up operation performed by the computer the top number will be the number along the side of the table. The bottom number will be the number along the top of the ta table in Figure 4-1.

$$\begin{array}{r} 1355 \\ \underline{5331} \\ 0068 \end{array}$$

A test is made for an overflow from this addition. There is an intersection between cube A and cube B since there is no overflow, and cube A will be reduced to a set of smaller cubes. (i.e. cube C and cube D) The operation continues by adding the transform of the first operation with a "1" to the right to a zero field.

$$\begin{array}{r} 00000 \\ \underline{00681} \\ 0009\overline{0} \end{array}$$

This result is checked for an overflow. Since there is no overflow the result is used for a further operation by adding it to cube A.

$$\begin{array}{r} 1355 \\ \underline{0009} \\ 1353 \end{array}$$

This gives cube C the first cube of the answer. The process continues with a clearing operation to the transform.

$$\begin{array}{r} 0068 \\ \underline{0009} \\ 0060 \end{array}$$

This number becomes the new transform and the process is

repeated from the first transform by adding the second transform with a "1" to the right to the first conversion.

$$\begin{array}{r} 00090 \\ 00601 \\ \hline 00700 \end{array}$$

The second conversion is checked for an overflow. As there is no overflow the operation continues by adding to cube A.

$$\begin{array}{r} 1355 \\ 0070 \\ \hline 1315 \end{array}$$

This gives cube D the second cube of the answer. The process now repeats for a third time by clearing the second transform by adding the second conversion.

$$\begin{array}{r} 0060 \\ 0070 \\ \hline 0000 \end{array}$$

The third conversion is formed by adding the third transform with a "1" to the right to the second conversion.

$$\begin{array}{r} 00700 \\ 00001 \\ \hline 00000 \end{array}$$

There is a carry across the entire answer field signaling an overflow condition. The sharping subroutine uses this overflow to end the sharping operation. All of the cubes to be generated have been generated by this point.

For an example of two cubes which do not intersect take these two cubes.

cube A = 01-- cube B = 1-01

The test will proceed by adding cube A to cube B to get the

transform. In this transform there is an overflow on the test indicating that cube A and cube B do not intersect.

$$\begin{array}{r} 1355 \\ \underline{3513} \\ \overline{0}086 \end{array}$$

To sharp cube B from cube A would be to remove nothing from cube A leaving it unaltered as the answer, and the sharping subroutine does just that.

For the subsuming operation the compare operation code is used. The compare on a 1620 is similar to the subtract operation code with the exception that the result is not entered into storage. After a compare operation the result can be tested from the condition of the High/Positive and Equal/Zero indicators. In the subtraction process the digit from the subtrahend which makes up the units position in the table look-up address is complemented. For the units position of the data the ten's complement is used. For every position after that the ten's complement is used if the preceeding result had a flag on the number found in the table. If no flag was found the nine's complement is used. To achieve the subsuming function the add tables have been overlayed with the table in Figure 4-2. Again all the unfilled positions in this table are invalid combinations for the subsuming operation. The fact that the number two is not used in the cube representation is of no importance. The important factor is that the result is not zero. The operation of the flag will serve to indicate if the compare is high or low. In a com-

pare operation in which the result is non-zero the High/Positive indicator is turned on if the result of the last subtraction has a flag over it.

	0	1	2	3	4	5	6	7	8	9
1									2	$\overline{2}$
3							2	$\overline{2}$		
5					2	$\overline{2}$	2	$\overline{2}$	2	$\overline{2}$

FIGURE 4-2

Inspection of two cubes of which one covers the other will show that every literal in the small cube matches every literal in the larger cube or the larger cube will have an unspecified literal where the smaller cube has a "1" or "0" literal. For example cube A covers cube B, but does not cover cube C or cube D.

cube A = --1-0- cube C = -1010-

cube B = -11-01 cube D = -1-10-

In the first case cube B is compared to cube A. The

$$\begin{array}{r} 553515 \\ \underline{533513} \\ \overline{222222} \end{array}$$

carry in the left hand position turns on the High/Positive indicator which is the signal to the subsuming subroutine

that cube A covers cube B. In the second case cube C is compared to cube A.

553515
531315

222222

In the fourth position from the right the comparison of one to three causes a no flag result. This causes the nine's compliment of the lower number to be used in the fifth position. An inspection of the add table will show that this no flag condition will be propagated for the rest of the operation, and the result will test low signaling that cube A does not cover cube C. In the third case cube D is compared to cube A and the result will be the same as the second case.

553515
535315

222222

The relationship of these two subroutines to the entire program is shown in the next chapter.

CHAPTER V

FLOW CHARTS OF THE PROGRAMS

The flow charts of the two level minimization program on the IBM 1620 are presented in this chapter to show the organization of the program.

The first flow chart Figure 5-1 shows the organization of the main program. An inspection of the chart will show that the main program is concerned with the task of allocating the storage area for the cubical arrays and controlling the operations to be performed on the arrays. The actual operations on the cubical arrays are performed by subroutines.

The major sharping subroutine is given in Figure 5-2. The sharping subroutine is concerned with the operation of sharping one cubical array from a second array as dictated by the main program. The sharping subroutine uses two subroutines to aid it in its operation. One subroutine is the single sharp subroutine given in Figure 5-3. The subroutine performs the operation of sharping a single cube from another cube by the method explained in Chapter IV. The second subroutine used by the major sharping subroutine is the subsuming subroutine given in Figure 5-4. This subroutine performs

TWO LEVEL MINIMIZATION

To minimize a Boolean expression into two level logic

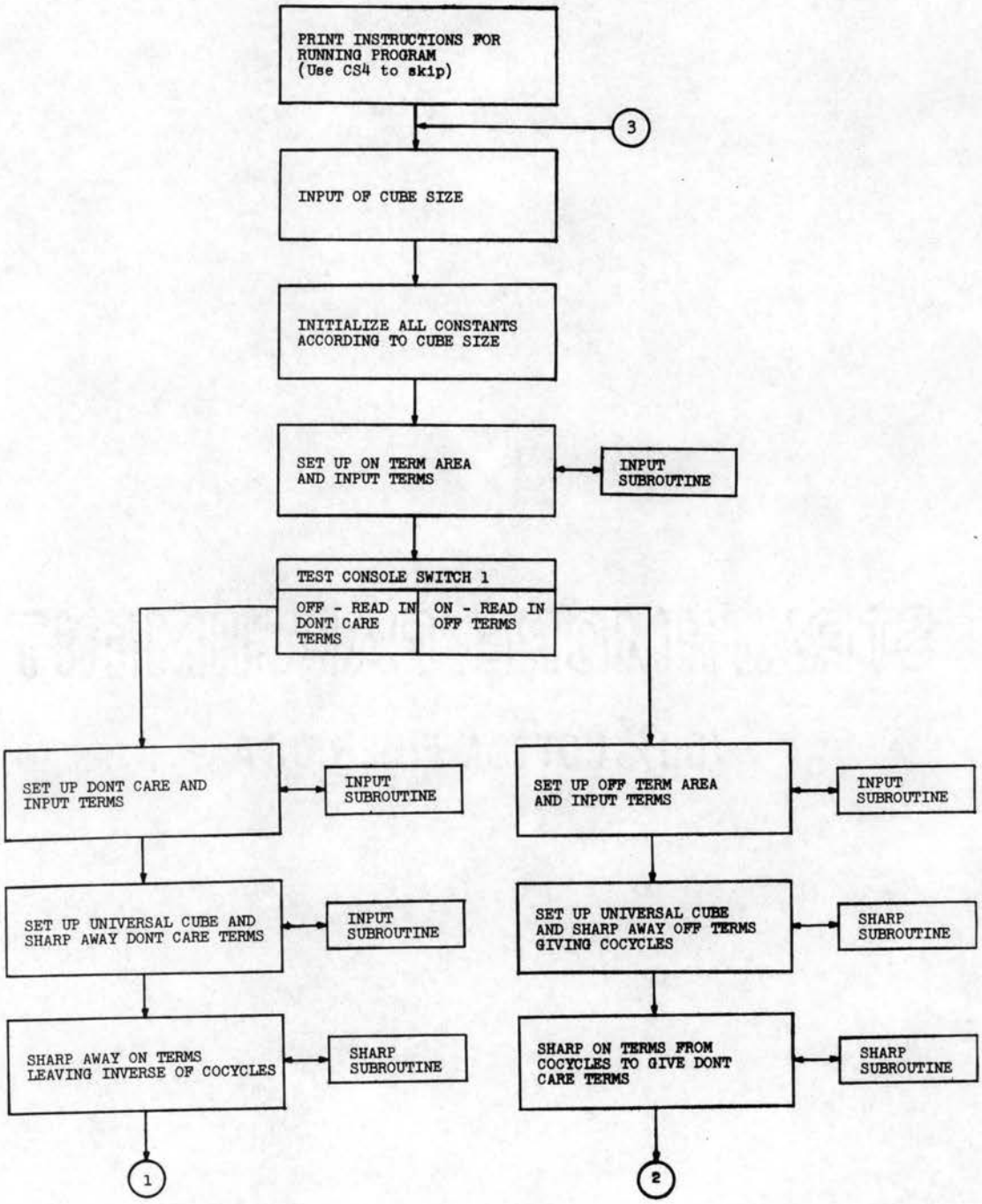

FIGURE 5-1

To continue Figure 5-1

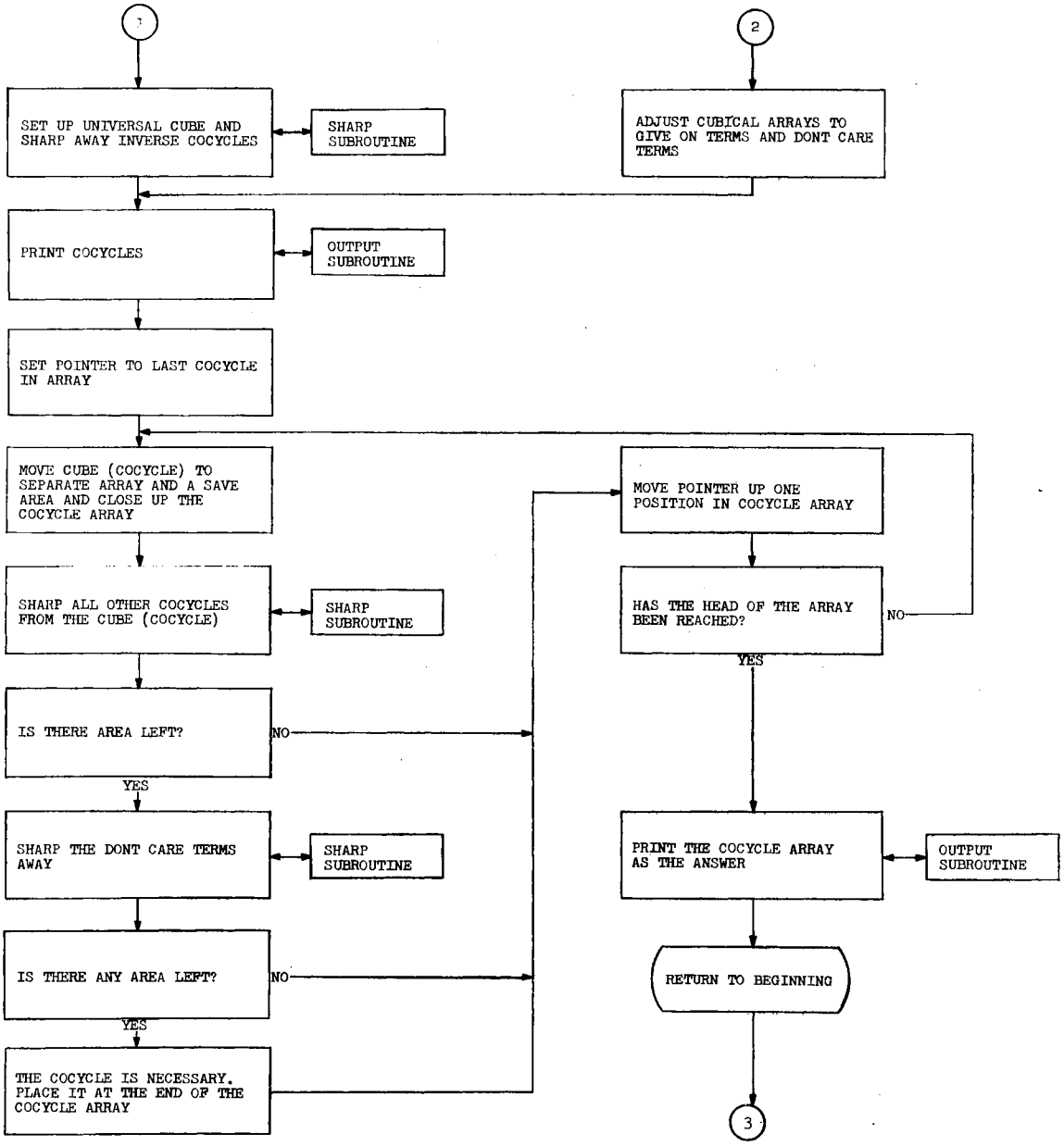

FIGURE 5-1

SHARPING SUBROUTINE

To sharp the cubical array RRPL from the cubical array TABL

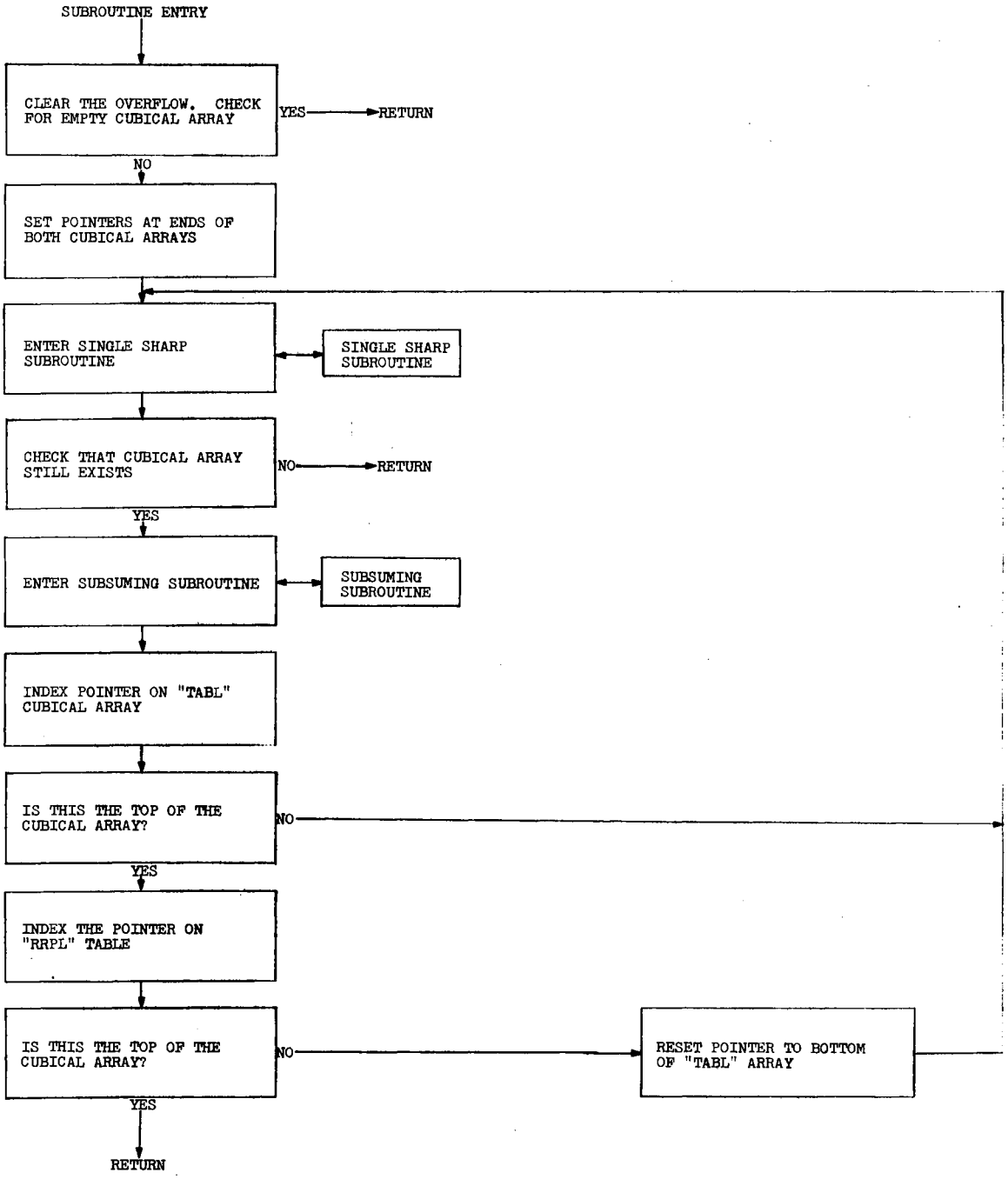

FIGURE 5-2

SINGLE SHARP SUBROUTINE

To sharp a single cube TERM from a single cube TAB

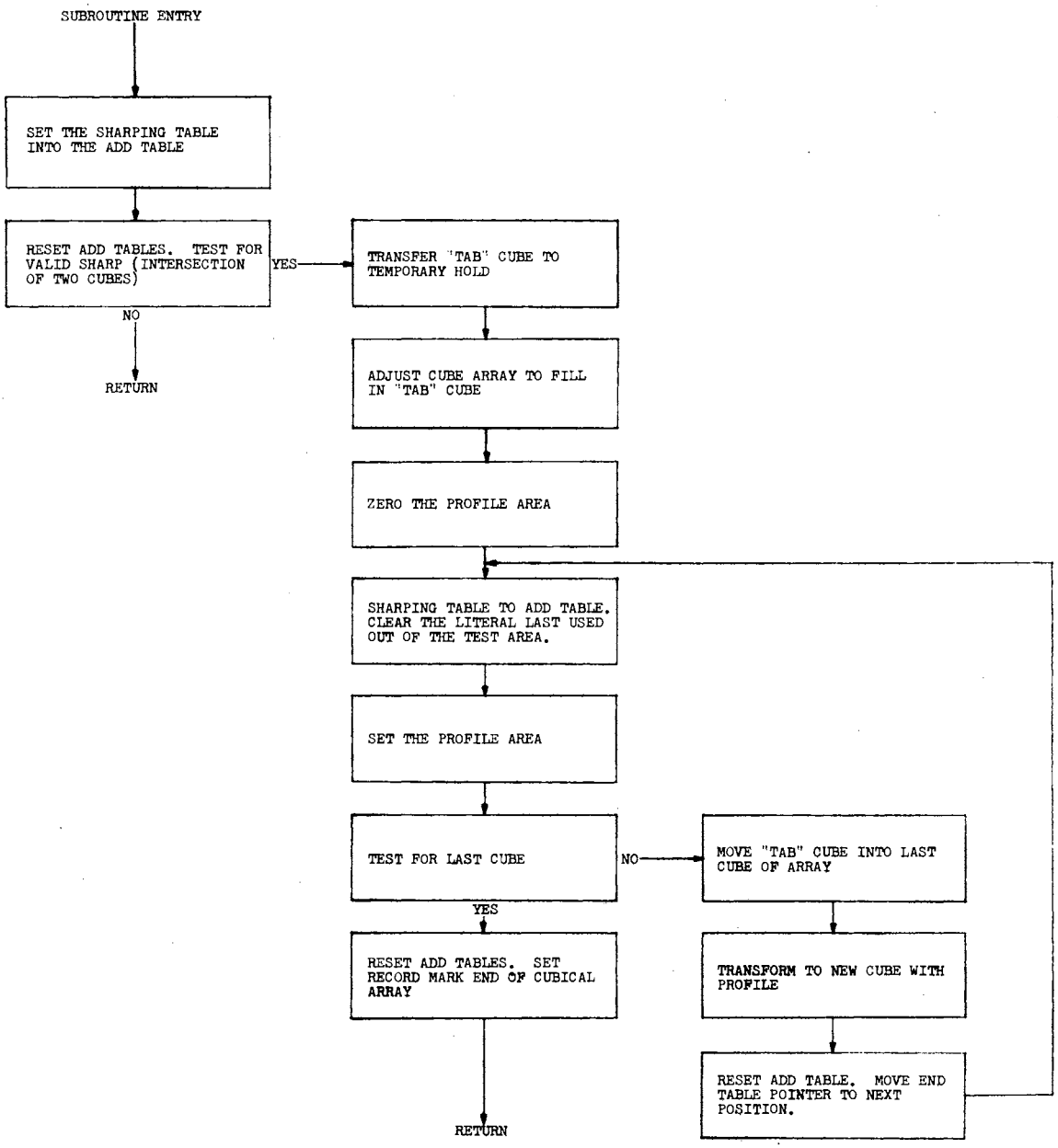

FIGURE 5-3

SUBSUMING SUBROUTINE

To subsume the single sharp cubes from the rest of the array

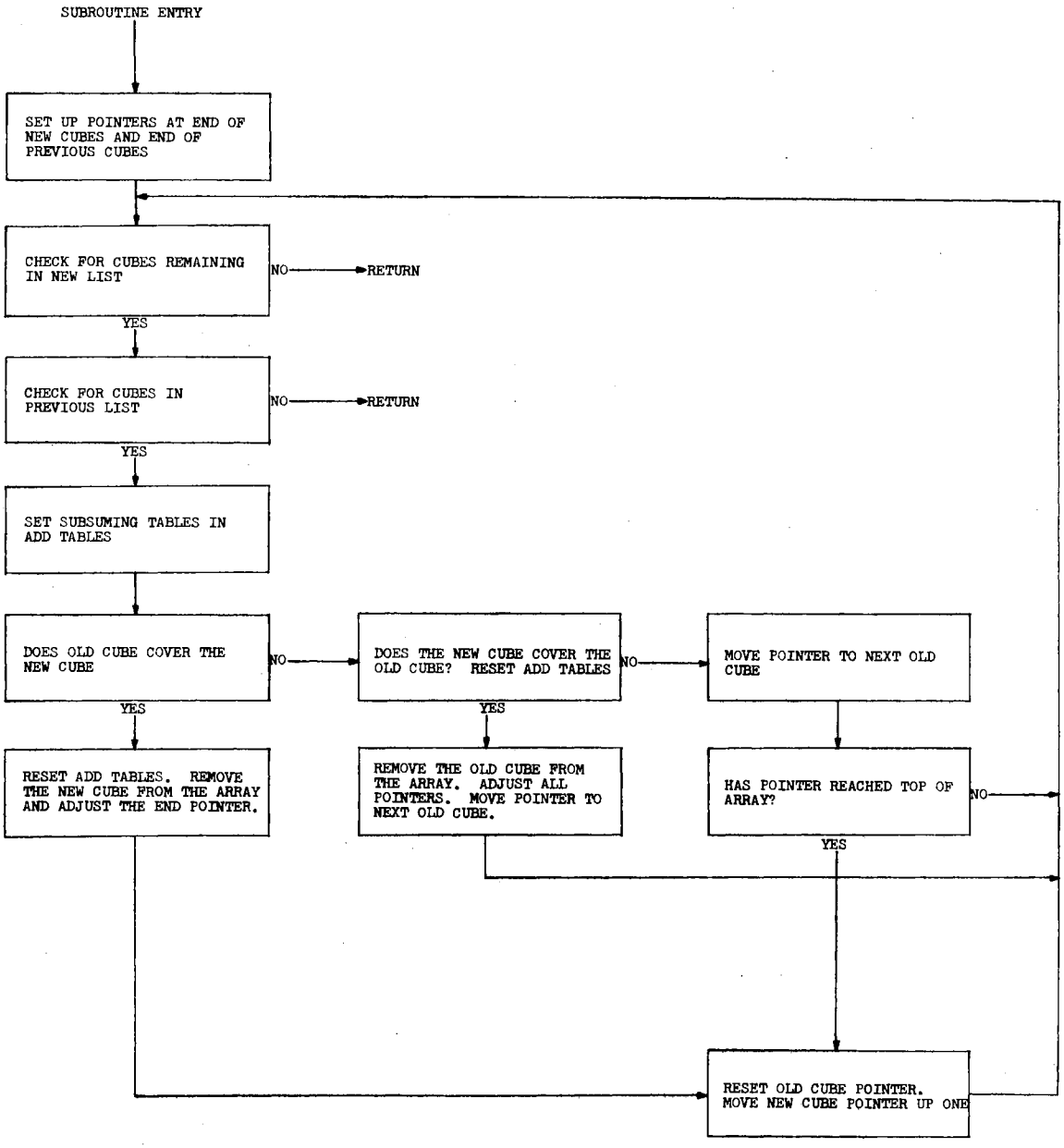

FIGURE 5-4

INPUT SUBROUTINE

To read cubes into AREA in the internal format

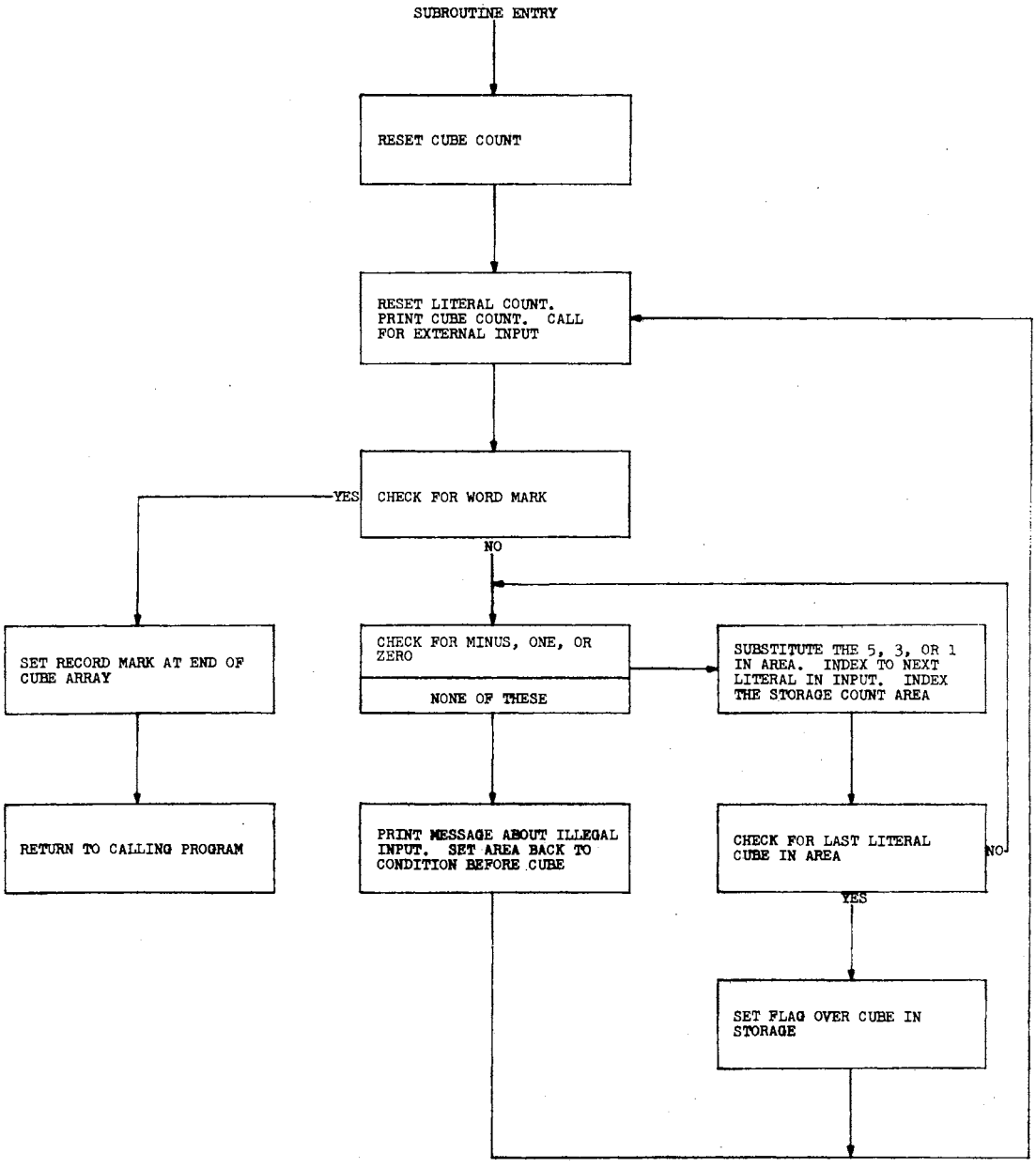

FIGURE 5-5

OUTPUT SUBROUTINE

To print out cubical array AREA

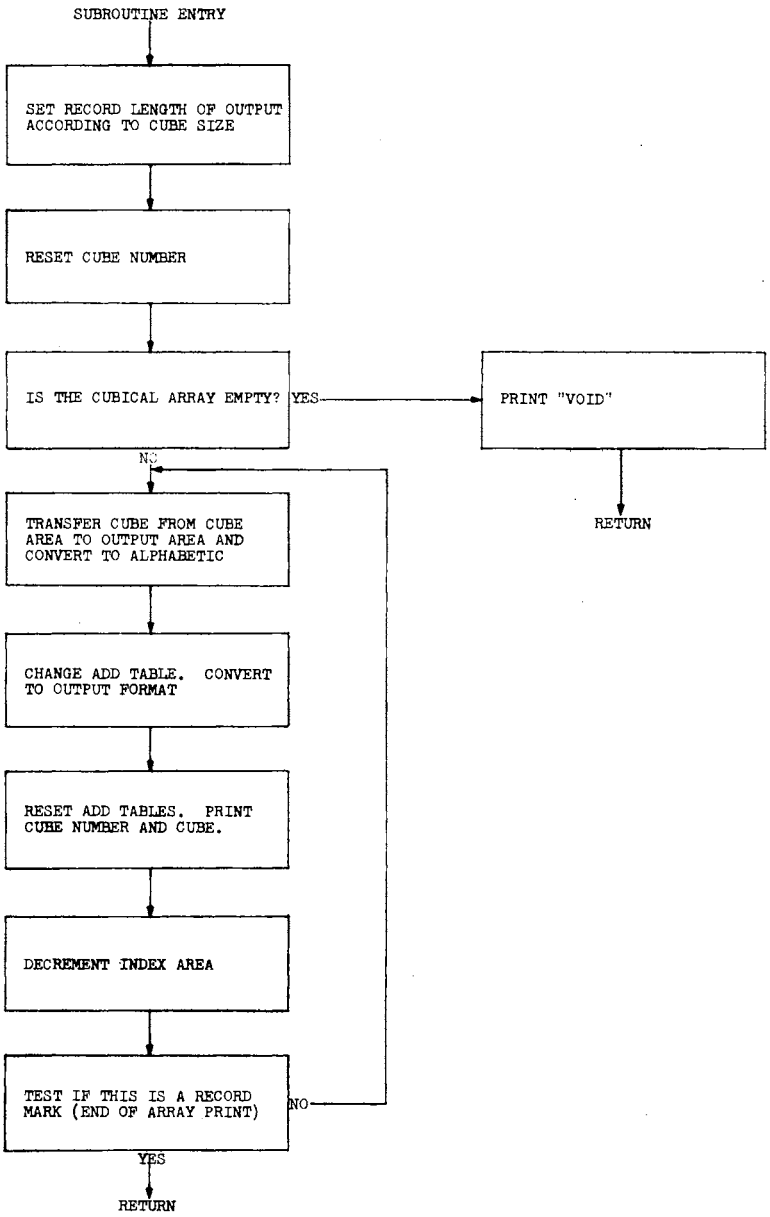

FIGURE 5-6

the subsuming operation between the last part of a cubical array and the first part. The cubical array and the division point between the two parts is given by the major sharping subroutine. In this program the subsuming subroutine is used to check the cubes generated by the single sharp subroutine to see that they are not covered by any cubes already in the array.

The input subroutine given in figure 5-5 performs the function of accepting cubes from a person at the input keyboard and converting them to the internal format which is used in the program.

The output subroutine given in figure 5-6 performs the function of printing a cubical array on the output typewriter.

If one wished to pursue this logic process further he could use these subroutines to do the logic operations. The main program would have to be rewritten. The main program is by its very nature of directing the process of the operations peculiar to the individual problem.

CHAPTER VI

EXAMPLES OF PROBLEMS

To illustrate the operation of the program sample problems have been selected in which the answers can also be shown through the use of maps or by reasoning.

The first example is a four literal problem which illustrates the order dependance of the forced irredundant cover method.

The ON TERMS are:

1. 0100
2. 1100

The OFF TERMS are:

1. 0001
2. 0111
3. 1010

The COCYCLES ARE:

1. 1-0-	5. -011
2. -10-	6. 1--1
3. --00	7. 11--
4. 001-	8. -1-0
9. 0--0	

The map in Figure 6-1 shows the on terms marked with "N" and the off terms marked with "F". The cocycles are shown by a rectangle or a line drawn through the area which they occupy. Lines drawn out one side of the map are considered continuous with the lines drawn out the opposite side of the

map. There are three cocycles each of which covers the on terms. They are:

2. -10-
3. --00
8. -1-0

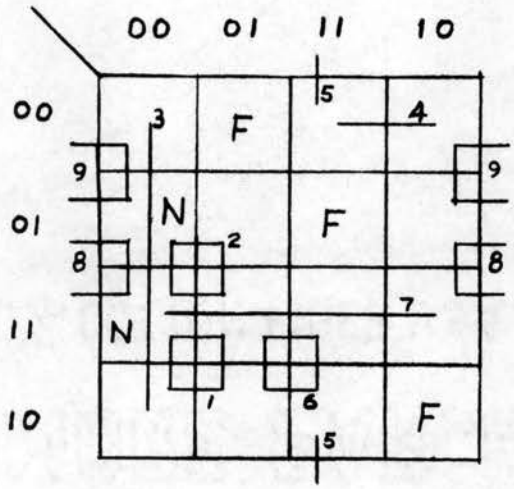

FIGURE 6-1

When the on and off terms are input to the computer in the order shown in Figure 6-2 the eighth cocycle in the list is the last cube considered which still covers the on terms forcing it to be the answer. All the preceding cubes are considered to be redundant.

When the on and off terms are ordered as in Figure 6-3 the cocycles take on a new order. The second cube from the Figure 6-2 becomes the last cube considered, and therefore, the answer. A third ordering of the on and off terms in Figure 6-4 gives still another ordering of cocycles. However, the eight cube from Figure 6-2 is still the chosen answer.

LIST OF ON TERMS	LIST OF COCYCLES
1. 0100	1. 1-0-
2. 1100	2. -10-
	3. --00
	4. 001-
LIST OF OFF TERMS	5. -011
	6. 1--1
1. 0001	7. 11--
2. 0111	8. -1-0
3. 1010	9. 0--0

MINIMIZED SOLUTION

1. -1-0

FIGURE 6-2

LIST OF ON TERMS	LIST OF COCYCLES
1. 0100	1. 0--0
2. 1100	2. -1-0
	3. --00
LIST OF OFF TERMS	4. 11--
	5. 1-0-
1. 1010	6. 1--1
2. 0111	7. 001-
3. 0001	8. -011
	9. -10-

MINIMIZED SOLUTION

1. -10-

FIGURE 6-3

The second example is take from a decimal half adder which produces the sum of two decimal digits. The decimal digits are represented in binary coded decimal Figure 6-5. The function to be found is the relation of the lower bit to the input code. A review of the list of binary coded

LIST OF ON TERMS	LIST OF COCYCLES
1. 0100	1. 1-0-
2. 1100	2. -10-
	3. --00
LIST OF OFF TERMS	4. -011
	5. 001-
1. 0001	6. 0--0
2. 1010	7. -1-0
3. 0111	8. 11--
	9. 1--1

MINIMIZED SOLUTION

1. -1-0

FIGURE 6-4

0 = 0000	5 = 0101
1 = 0001	6 = 0110
2 = 0010	7 = 0111
3 = 0011	8 = 1000
4 = 0100	9 = 1001

FIGURE 6-5

decimal will show that this bit is "1" in the cases where the sum is 1, 3, 5, 7, or 9. Therefore, the on terms will be all pairs of digits which sum to these numbers. There are eight inputs to the decimal half adder. The first four are the first decimal digit, and the second four are the second decimal digit. The off terms will be pairs of digits which sum to 0, 2, 4, 6, or 8. These inputs are shown in Figure 6-6. Inspection shows that to produce an odd sum requires that an even number be added to an odd number and

LIST OF ON TERMS

1. 00000001	18. 01110110	35. 01000011
2. 00010000	19. 10000101	36. 01010010
3. 00101001	20. 10010100	37. 01100001
4. 00111000	21. 00000101	38. 01110000
5. 01000111	22. 00010100	39. 10001001
6. 01010110	23. 00100011	40. 10011000
7. 01100101	24. 00110010	41. 00001001
8. 01110100	25. 01000001	42. 00011000
9. 10000011	26. 01010000	43. 00100111
10. 10010010	27. 01101001	44. 00110110
11. 00000011	28. 01111000	45. 01000101
12. 00010010	29. 10000111	46. 01010100
13 .00100001	30. 10010110	47. 01100011
14. 00110000	31. 00000111	48. 01110010
15. 01001001	32. 00010110	49. 10000001
16. 01011000	33. 00100101	50. 10010000
17. 01100111	34. 00110100	

LIST OF OFF TERMS

1. 00000000	18. 01110101	35. 01000010
2. 00011001	19. 10000100	36. 01010001
3. 00101000	20. 10010011	37. 01100000
4. 00110111	21. 00000100	38. 01111001
5. 01001000	22. 00010011	39. 10001000
6. 01010111	23. 00100010	40. 10010111
7. 01100110	24. 00110001	41. 00001000
8. 01110101	25. 01000000	42. 00010111
9. 10000100	26. 01011001	43. 00100110
10. 10010011	27. 01101000	44. 00110101
11. 00000010	28. 01110111	45. 01000100
12. 00010001	29. 10000110	46. 01010011
13. 00100000	30. 10010101	47. 01100010
14. 00111001	31. 00000110	48. 01110001
15. 01001000	32. 00010101	49. 10000000
16. 01010111	33. 00100100	50. 10011001
17. 01100110	34. 00110011	

LIST OF COCYCLES

1. ----11--
2. ----1-1-
3. 1-1-----
4. 11------
5. ---1---0
6. ---0---1

MINIMIZED SOLUTION

1. ---0---1
2. ---1---0

FIGURE 6-6

the reverse also an odd number added to an even number. Further inspection shows that the outcome of the lower order bit depends only on the lower order bit of the two digits being summed. Also when the lower order bit of one digit is one the lower order bit of the other is zero and the reverse. This is indeed the conclusion that the minimization program reaches giving the answer.

1. ---0---1
2. ---1---0

Of interest to the reader would be the running times for these examples. The four literal problem takes approximately 10 seconds to run. The eight literal problem takes approximately 10 minutes to run plus the time taken to type the input. One would not want to run such an obvious case, but there are more difficult examples to be found.

CHAPTER VII

SUMMARY AND CONCLUSIONS

For this study a two level minimization program has been written which anyone with access to an IBM 1620 Model I can run. The program is written in subroutine form making it easy to modify. In the process of writing this program the add tables in addressable core storage of the 1620 were found to be very useful.

When the program was started it seemed that core storage would limit the number of literals that could be used in a given problem run. As the program now stands running time would become excessive before core storage was exhausted. The program is now built to take up to ten literals. At the present eight to nine literals is about the limit.

The author has found that while changing the add tables of the 1620 to perform unusual functions can save considerable time there is also a penalty. The penalty is this. If one is performing this special function intermixed with with arithmetical operations in rapid succession there is a considerable amount of computer time used in moving in the special add tables for the special function and then moving in the regular add tables for arithmetical operations. If

one can arrange to separate these different types of operations an even greater savings of computer time can be achieved.

There is an addition to this problem which might make an interesting study, and it is of about equal magnitude to the program just completed. This is in the form of the input. As the program stands, a person using it has to make up the on and off terms. For some types of logic this is not difficult. But, there is a translate function in which one could make a list of combinations and a corresponding list of combinations that the first combinations should translate into. Just to strip out the on and off terms from such a list would be a tedious chore in itself. A program could be written to take each literal from the second set of combinations and make up the on and off terms for the literal. Then use the two level minimization program to solve for the function for each literal. As these functions tend to run from five to eight literals, I feel that there is still room in the core storage if one plans a good overlay scheme in which he reuses areas of storage for other functions after he has used them for the translate function.

BIBLIOGRAPHY

1. Caldwell, S. H., Switching Circuits and Logical Design New York: John Wiley and Sons, Inc., 1959, pp 124-32

2. Marcus, M.P., Switching Circuits for Engineers, Englewood Cliffs, N. J., Prentice-Hall, Inc., pp 85-99

3. Staff of the Computation Laboratory, Synthesis of Electronic Computing and Control Circuits, Cambridge, Mass. Harvard University Press

4. Quine, W. V., "The Problem of Simplifying Truth-Functions." The American Mathematical Monthly, LIX (Oct, 1952) 521-31

5. Roth, J. P., "Algebraic Topological Methods for the Synthesis of Switching Systems I," Transactions of the American Mathematical Society, Vol. 88, No. 2, July 1958 pp 301-326

6. Kircher, T. A., Lecture Notes for a Course on Advanced Logic Design Techniques, IBM Development Laboratory, Endicott, New York

7. McFarlin, F. E., "A Technique for Minimizing Boolean Functions That Does Not Require a Canonical Form." (pamphlet proposed for publication in the IRE Transactions on Electronic Computers, Endicott, New York December 1958)

8. Vipraio, W. J., Determination of Prime Implicants for Disjunctive Boolean Functions by use of a Digital Computer,(unpub. Masters Thesis, Oklahoma State University, 1960)

9. Petrick, S. R., A Direct Determination of the Irredundant Forms of a Boolean Function from the Set of Prime Implicants. USAF Cambridge Research Center TR-56-110 (April, 1956)

10. Reference Manual, IBM 1620 Data Processing System, (International Business Machines Corporation, 1961) A26-4500

APPENDIX A

Instructions for running the Two Level Minimization program.

1. The Two Level Minimization program is a self loading program for the IBM 1620 Model I. (The add tables must be in memory. This is not the case for a Model II.) Set up the machine to load cards, ready the cards in the card reader, and load.

2. The program prints instructions for its running on the console typewriter. These instructions can be suppressed by using console switch 4. (In general, console switch 4 is used for skipping purposes.)

3. The program will ask for the cube size. The input response should be 5 digits with leading zeroes on the console typewriter. The value should be between one and ten. Use release-start after the digits.

4. A choice between the dont care array or the off array should be made by this point. Use console switch 1 for the off array.

5. The program will give the title for the on array and the first number. The cubes are given one at a time followed by the release-start key. The cubes will consist of "1's", "0's", and "-'s". After each cube the program will return for another cube. After all of the cubes have been entered, enter one more cube consisting of a record mark followed by release-start to signal the end of the array. If a mistake is made in a cube while typing, turn on console switch 4 before using release-start. The program will ignore the input and return for the same cube agin. The skip on console switch 4 is effective regardless of the number of characters which have been entered.

6. The program will give the title for the dont care or the off array depending on the setting of the console switch 1. Give the cubes in the same manner as the on array.

7. After the record mark, release-start on the dont care or off array the program will calculate the cocycles and print the list. This printing can be suppressed with console switch 4.

8. The program will then calculate a minimized two level solution, and print out the list of cubes as the answer to the run. After these answers the program starts another problem by asking for the cube size. Repeat the instructions from step 3 if another run is desired.

9. When the program is in core storage it can be restarted after an unusual stop by transferring to location 600.

```
         DORG  402                                                      00402
CUBE     DC    5,0,,        LOCATION OF ON TERM LIST                  00406 00005
CUBEL    DC    5,0,,        LOCATION OF LAST TERM IN ON TERM LIST     00411 00005
DONT     DC    5,0,,            DONT CARE LIST                        00416 00005
DONTL    DC    5,0,,            LAST TERM IN DONT CARE LIST           00421 00005
SAVE     DC    5,0,,        AREA FOR SAVING LISTS                     00426 00005
SAVEL    DC    5,0                                                    00431 00005
WORK     DC    5,0,,        GENERAL WORK AREA LIST                    00436 00005
WORKL    DC    5,0                                                    00441 00005
AREA     DC    5,0,,        LOCATION OF INPUT OUTPUT AREAS            00446 00005
RRPL     DC    5,0,,        LOCATION OF LIST TO DO SHARPING WITH      00451 00005
TERM     DC    5,0,,        INDEX USED IN SHARPGIN OPERATION          00456 00005
TABL     DC    5,0,,        LOCATION OF LIST TO DO SHARPING TO        00461 00005
TABE     DC    5,0                                                    00466 00005
TAB      DC    5,0,,        *                                         00471 00005
TASS     DC    5,0,,        INDEX USED IN SUBSUMING OPERATION         00476 00005
TAS      DC    5,0,,        *                                         00481 00005
TSSS     DC    5,0                                                    00486 00005
TSS      DC    5,0,,        *                                         00491 00005
FIRC     DC    5,0,,        INDEX USED IN FINDING IRREDUNDANT COVER   00496 00005
ZSOL     DSA   ZS1                                                    00501 00005 -0549
UNIV     DSA   UN1                                                    00506 00005 -0560
N        DS    5                                                      00511 00005
N1       DS    5                                                      00516 00005
TEST     DS    10                                                     00526 00010
         DS    1                                                      00527 00001
SOL      DS    10                                                     00537 00010
         DS    1                                                      00538 00001
HOLD     DS    10                                                     00548 00010
ZS1      DC    1,0                                                    00549 00001
         DSC   10,0                                                   00550 00010
UN1      DS    1,                                                     00560 00001
         DC    10,5555555555                                          00570 00010
STORE    DS    10                                                     00580 00010
TWO      DC    5,2                                                    00585 00005
ONE      DC    5,1                                                    00590 00005
ZERO     DC    5,0                                                    00595 00005
         DC    1,@                                                    00596 00001
         DORG  600                                                    00600
* MAIN PROGRAM STRING
TLM      TR    TABLE,ADDTB                                            00600 31 00300 00130
```

```
          RCTY                                                                00612 34 00000 00102
          WATY NOTE1                                                          00624 39 04739 00100
          RCTY                                                                00636 34 00000 00102
          RNTY INPUT-1,,,          DEFINE SIZE OF CUBES TO BE USED - OUTSIDE  00648 36 00024 00100
          RCTY                                                                00660 34 00000 00102
          SF   INPUT-1                                                        00672 32 00024 00000
          TF   N,INPUT+3                                                      00684 26 00511 00028
          TF   N1,N,,              DEVELOP CONSTANT ONE LESS THAN SIZE OF CU  00696 26 00516 00511
          S    N1,ONE                                                         00708 22 00516 00590
          TFM  ZSOL,ZS1,,          ADJUST THE SIZE OF THESE CONSTANTS TO SIZ  00720 16 00501 -0549
          A    ZSOL,N,,            CUBE                                      00732 21 00501 00511
          TFM  UNIV,UN1                                                       00744 16 00506 -0560
          A    UNIV,N                                                         00756 21 00506 00511
          TFM  CUBE,FREE                                                      00768 16 00406 -5127
          TF   AREA,CUBE,,         SET UP ON TERM AREA                        00780 26 00446 00406
          TFM  INX,TLM1                                                       00792 16 03064 -0852
          RCTY                                                                00804 34 00000 00102
          WATY TNTN1                                                          00816 39 04947 00100
          RCTY                                                                00828 34 00000 00102
          B    INN,,,              INPUT THE ON CUBES                         00840 49 02556 00000
TLM1      TF   CUBEL,AREA                                                     00852 26 00411 00446
          RCTY                                                                00864 34 00000 00102
          BC1  TLM3,,,             IF CONSOLE SWITCH 3 GO TO OFF TERMS        00876 46 01992 00100
          TF   DONT,CUBEL,,        SET-UP DONT CARE AREA                      00888 26 00416 00411
          A    DONT,TWO                                                       00900 21 00416 00585
          TF   AREA,DONT                                                      00912 26 00446 00416
          TFM  INX,TLM2                                                       00924 16 03064 -0984
          RCTY                                                                00936 34 00000 00102
          WATY TNTN2                                                          00948 39 04981 00100
          RCTY                                                                00960 34 00000 00102
          B    INN,,,              INPUT THE DONT CARE TERMS                  00972 49 02556 00000
TLM2      TF   DONTL,AREA                                                     00984 26 00421 00446
          RCTY                                                                00996 34 00000 00102
          TF   SAVE,DONTL                                                     01008 26 00426 00421
          A    SAVE,TWO                                                       01020 21 00426 00585
          TF   WORK,SAVE,,         SET-UP WORK AREA FOR SHARPING AWAY THE     01032 26 00436 00426
TLM20     TF   WORKL,WORK,,        ON TERMS AND DONT CARE TERMS               01044 26 00441 00436
          A    WORKL,N                                                        01056 21 00441 00511
          TD   WORKL,RM,6                                                     01068 25 0044J 00401
          S    WORKL,ONE                                                      01080 22 00441 00590
          TF   WORKL,UNIV,611                                                 01092 26 0044J 00500
```

```
            TF    TABL,WORKL                                                01104 26 00461 00441
            TF    RRPL,DONTL                                                01116 26 00451 00421
            TFM   RX,TLM21                                                  01128 16 03745 -1164
            TFM   RX,TLM21                                                  01140 16 03745 -1164
            B     SHARP,,,          SHARP THE DONT CARE TERMS               01152 49 03494 00000
TLM21       TF    RRPL,CUBEL                                                01164 26 00451 00411
            TFM   RX,TLM22                                                  01176 16 03745 -1200
            B     SHARP,,,          SHARP THE ON TERMS                      01188 49 03494 00000
TLM22       TF    SAVEL,TABL                                                01200 26 00431 00461
            TF    WORK,SAVEL,,      SAVE THE RESULTS FOR THE RESHARPING OPERA 01212 26 00436 00431
            A     WORK,TWO                                                  01224 21 00436 00585
            TF    WORKL,WORK                                                01236 26 00441 00436
            A     WORKL,N                                                   01248 21 00441 00511
            TD    WORKL,RM,6                                                01260 25 0044J 00401
            S     WORKL,ONE                                                 01272 22 00441 00590
            TF    WORKL,UNIV,611                                            01284 26 0044J 00500
            TF    TABL,WORKL                                                01296 26 00461 00441
            TF    RRPL,SAVEL                                                01308 26 00451 00431
            TFM   RX,TLM23                                                  01320 16 03745 -1344
            B     SHARP,,,          RESHARP OPERATION TO GET COCYCLES        01332 49 03494 00000
TLM23       TF    SAVEL,TABL                                                01344 26 00431 00461
            S     SAVEL,WORK                                                01356 22 00431 00436
            A     SAVEL,SAVE                                                01368 21 00431 00426
            TR    SAVE,WORK,611                                             01380 31 00420 00430
TLM25       TF    AREA,SAVEL                                                01392 26 00446 00431
            RCTY                                                            01404 34 00000 00102
            WATY  NOTE2                                                     01416 39 04807 00100
            RCTY                                                            01428 34 00000 00102
            TFM   OUTX,TLM24,,      PRINT OUT OF COCYCLES                   01440 16 03466 -1464
            B     OUT                                                       01452 49 03102 00000
TLM24       TF    FIRC,SAVEL                                                01464 26 00496 00431
            BNR   *+24,FIRC,11                                              01476 45 01500 00490
            B     TLM60                                                     01488 49 01932 00000
            TF    WORK,SAVEL                                                01500 26 00436 00431
            A     WORK,TWO                                                  01512 21 00436 00585
TLM50       TF    WORKL,WORK,,      SET-UP WORK AREA FOR TESTING ONE COCYCLE 01524 26 00441 00436
            A     WORKL,N                                                   01536 21 00441 00511
            TD    WORKL,RM,6                                                01548 25 0044J 00401
            S     WORKL,ONE                                                 01560 22 00441 00590
            TF    STORE,FIRC,11,,   TRANSFER COCYCLE BEING CONSIDERED FOR TES 01572 26 00580 00490
            TF    TABL,WORKL                                                01584 26 00461 00441
```

```
          TF    TABL,FIRC,611                                                01596 26 0046J 00490
          S     SAVEL,N                                                      01608 22 00431 00511
          TF    TLM52+6,FIRC                                                 01620 26 01674 00496
          S     TLM52+6,N1                                                   01632 22 01674 00516
          TF    TLM52+11,FIRC                                                01644 26 01679 00496
          A     TLM52+11,ONE                                                 01656 21 01679 00590
TLM52     TR    0,0,,          CLOSE IN THE GAP LEFT BY COCYCLE UNDER CO     01668 31 00000 00000
          TF    RRPL,SAVEL                                                   01680 26 00451 00431
          TFM   RX,TLM51                                                     01692 16 03745 -1716
          B     SHARP,,,       SHARP AWAY ALL THE REMAINING COCYCLES         01704 49 03494 00000
TLM51     BNR   TLM53,TABL,11                                                01716 45 01740 0046J
          B     TLM56                                                        01728 49 01824 00000
TLM53     TF    RRPL,DONTL                                                   01740 26 00451 00421
          TFM   RX,TLM54                                                     01752 16 03745 -1776
          B     SHARP,,,       SHARP AWAY THE DONT CARE TERMS                01764 49 03494 00000
TLM54     BNR   TLM55,TABL,11                                                01776 45 01800 0046J
          B     TLM56                                                        01788 49 01824 00000
TLM55     A     SAVEL,N,,      COCYCLE IS NECESSARY - ADD IT BACK TO COC     01800 21 00431 00511
          TF    SAVEL,STORE,6, LIST                                         01812 26 0043J 00580
TLM56     S     FIRC,N,,       THE COCYCLE IS REDUNDANT - REMOVE AND FOR     01824 22 00496 00511
          BNR   TLM50,FIRC,11                                                01836 45 01524 00490
          RCTY                                                               01848 34 00000 00102
          RCTY                                                               01860 34 00000 00102
          WATY  TNTN3                                                        01872 39 05029 00100
          RCTY                                                               01884 34 00000 00102
          TF    AREA,SAVEL                                                   01896 26 00446 00431
          TFM   OUTX,TLM60                                                   01908 16 03466 -1932
          B     OUT,,,         OUTPUT FOR MINIMIZED TERMS                    01920 49 03102 00000
TLM60     RCTY                                                               01932 34 00000 00102
          RCTY                                                               01944 34 00000 00102
          WATY  NOTE4                                                        01956 39 04907 00100
          RCTY                                                               01968 34 00000 00102
          B     TLM,,,         END OF PROGRAM                                01980 49 00600 00000
TLM3      TF    SAVE,CUBEL,,   ALTERNATE PROCEEDURE - INPUT OF OFF TERMS     01992 26 00426 00411
          A     SAVE,TWO,,     RATHER THAN DONT CARE TERMS                   02004 21 00426 00585
          TF    AREA,SAVE                                                    02016 26 00446 00426
          TFM   INX,TLM4                                                     02028 16 03064 -2088
          RCTY                                                               02040 34 00000 00102
          WATY  TNTN4                                                        02052 39 05091 00100
          RCTY                                                               02064 34 00000 00102
          B     INN,,,         INPUT OFF TERMS                               02076 49 02556 00000
```

```
      TLM4   TF    SAVEL,AREA                                          02088 26 00431 00446
             RCTY                                                      02100 34 00000 00102
             TF    WORK,SAVEL                                          02112 26 00436 00431
             A     WORK,TWO                                            02124 21 00436 00585
             TF    WORKL,WORK                                          02136 26 00441 00436
             A     WORKL,N                                             02148 21 00441 00511
             TD    WORKL,RM,6                                          02160 25 0044J 00401
             S     WORKL,ONE                                           02172 22 00441 00590
             TF    WORKL,UNIV,611                                      02184 26 0044J 00500
             TF    TABL,WORKL                                          02196 26 00461 00441
             TF    RRPL,SAVEL                                          02208 26 00451 00431
             TFM   RX,TLM5                                             02220 16 03745 -2244
             B     SHARP,,,          SHARP OFF TERMS FROM UNIVERSAL CUBE TO  02232 49 03494 00000
      TLM5   TF    WORKL,TABL,,      OBTAIN CUCYCLES                     02244 26 00441 00461
             TF    DONT,WORKL                                          02256 26 00416 00441
             A     DONT,TWO                                            02268 21 00416 00585
             TR    DONT,WORK,611                                       02280 31 00410 00430
             TF    DONTL,WORKL                                         02292 26 00421 00441
             S     DONTL,WORK                                          02304 22 00421 00436
             A     DONTL,DONT                                          02316 21 00421 00416
             TF    TABL,DONTL                                          02328 26 00461 00421
             TF    RRPL,CUBEL                                          02340 26 00451 00411
             TFM   RX,TLM6                                             02352 16 03745 -2376
             B     SHARP,,,          SHARP ON TERMS FROM CUCYCLES TO GIVE DONT  02364 49 03494 00000
      TLM6   TF    DONTL,TABL,,      CARE TERMS                          02376 26 00421 00461
             TF    AREA,DONTL                                          02388 26 00446 00421
             A     AREA,TWO,,        ADJUST AREAS AND RETURN TO MAIN STREAM  02400 21 00446 00585
             TR    AREA,WORK,611                                       02412 31 00440 00430
             TR    CUBE,DONT,611                                       02424 31 00400 00410
             S     DONTL,DONT                                          02436 22 00421 00416
             A     DONTL,CUBE                                          02448 21 00421 00406
             TF    DONT,CUBE                                           02460 26 00416 00406
             TF    SAVE,DONTL                                          02472 26 00426 00421
             A     SAVE,TWO                                            02484 21 00426 00585
             TR    SAVE,AREA,611                                       02496 31 00420 00440
             TF    SAVEL,WORKL                                         02508 26 00431 00441
             S     SAVEL,WORK                                          02520 22 00431 00436
             A     SAVEL,SAVE                                          02532 21 00431 00426
             B     TLM25,,,          RETURN TO MAIN STREAM             02544 49 01392 00000
     * INPUT SUBROUTINE FOR CUBES
      INN    S     TEMM,TEMM                                           02556 22 03088 03088
```

```
          A     TEMM,ONE,,          TRANSFORM CUBES FROM 0, 1, - TO 1, 3, 5     02568 21 03088 00590
          B     IN2                                                             02580 49 02988 00000
IN3       RATY  INPUT,,,            EXTERNAL INPUT                              02592 37 00025 00100
          BC4   IN2,,,              USE CONSOLE SWITCH 4 TO BYPASS A TERM       02604 46 02988 00400
          BNR   IN1,INPUT,,         WHICH HAS BEEN MISTYPED                     02616 45 02664 00025
          TD    AREA,RM,6,          RECORD MARK PRESENT - LAST TERM HAS BEEN    02628 25 00440 00401
          S     AREA,ONE,,          STORE RECORD MARK AT END OF ARRAY AND RET   02640 22 00446 00590
          B     INX,,6                                                          02652 49 0306M 00000
IN1       SF    IN5,,6                                                          02664 32 0307M 00000
          A     IN5,ONE                                                         02676 21 03074 00590
          C     IN5,LC5,6,          TEST FOR DONT CARE                          02688 24 0307M 03081
          BNE   IN10                                                            02700 47 02736 01200
          TDM   AREA,5,6                                                        02712 15 00440 00005
          B     IN13                                                            02724 49 02868 00000
IN10      C     IN5,LC3,6,          TEST FOR 1 VALUE                            02736 24 0307M 03083
          BNE   IN11                                                            02748 47 02784 01200
          TDM   AREA,3,6                                                        02760 15 00440 00003
          B     IN13                                                            02772 49 02868 00000
IN11      C     IN5,LC1,6,          TEST FOR 0 VALUE                            02784 24 0307M 03085
          BNE   IN12                                                            02796 47 02832 01200
          TDM   AREA,1,6                                                        02808 15 00440 00001
          B     IN13                                                            02820 49 02868 00000
IN12      WATY  NOTE3,,,            ERROR ON INPUT - UNEXPECTED VALUES          02832 39 04841 00100
          S     AREA,COUNT,,        RESET AND TRY AGAIN                         02844 22 00446 03100
          B     IN2                                                             02856 49 02988 00000
IN13      A     COUNT,ONE                                                       02868 21 03100 00590
          A     IN5,ONE                                                         02880 21 03074 00590
          A     AREA,ONE                                                        02892 21 00446 00590
          C     COUNT,N,,           TEST FOR END OF CUBE                        02904 24 03100 00511
          BNE   IN1                                                             02916 47 02664 01200
          S     AREA,N                                                          02928 22 00446 00511
          SF    AREA,,6,            SET THE FLAG FOR THIS TERM                  02940 32 00440 00000
          A     AREA,N                                                          02952 21 00446 00511
          A     TEMM,ONE                                                        02964 21 03088 00590
          BV    IN2                                                             02976 46 02988 01400
IN2       TF    COUNT,ZERO,,        RESET THE COUNTER                           02988 26 03100 00595
          TF    IN5,IN4                                                         03000 26 03074 03069
          RCTY                                                                  03012 34 00000 00102
          WNTY  TEMM-1,,,           NUMBER THE NEXT INPUT TERM                  03024 38 03087 00100
          WATY  TMM                                                             03036 39 03091 00100
          B     IN3                                                             03048 49 02592 00000
```

```
      INX     DS    5                                                    03064 00005
      IN4     DSA   INP                                                  03069 00005 -0024
      IN5     DS    5,                                                   03074 00005
      IN6     DS    5                                                    03079 00005
      LC5     DC    2,20                                                 03081 00002
      LC3     DC    2,71                                                 03083 00002
      LC1     DC    2,70                                                 03085 00002
      TEMM    DC    3,000                                                03088 00003
              DC    1,@                                                  03089 00001
      TMM     DAC   3,. @                                                03091 00006
      INP     DS    1,24                                                 00024 00001
      INPUT   DAS   50,25                                                00025 00100
      COUNT   DS    5                                                    03100 00005
     * OUTPUT SUBROUTINE FOR CUBES
      OUT     TFM   OUTC2,OUTPUT,,     PRINT OUT LIST GIVEN IN AREA      03102 16 03471 -3473
              A     OUTC2,N                                              03114 21 03471 00511
              A     OUTC2,N                                              03126 21 03471 00511
              TD    OUTC2,RM,6                                           03138 25 0347J 00401
              S     OUTC2,ONE                                            03150 22 03471 00590
              TD    OUTC2,RM,6                                           03162 25 0347J 00401
              S     OUTC2,ONE                                            03174 22 03471 00590
              S     TEMM,TEMM                                            03186 22 03088 03088
              BNR   OUT1,AREA,11,      IF THE CUBICAL ARRAY IS EMPTY PRINT 03198 45 03258 00440
              RCTY  ,,,                VOID                              03210 34 00000 00102
              WATY  NOTES                                                03222 39 04937 00100
              RCTY                                                       03234 34 00000 00102
              B     OUTX,,6                                              03246 49 03460 00000
      OUT1    TNF   OUTC2,AREA,611                                       03258 73 0347J 00440
              SF    OUTPUT-1                                             03270 32 03472 00000
              TDM   311,0                                                03282 15 00311 00000
              TDM   333,1                                                03294 15 00333 00001
              TDM   377,7                                                03306 15 00377 00007
              TDM   378,2                                                03318 15 00378 00002
              A     OUTC2,OUTC2,611                                      03330 21 0347J 0347J
              TR    TABLE,ADDTB                                          03342 31 00300 00130
              RCTY                                                       03354 34 00000 00102
              A     TEMM,ONE                                             03366 21 03088 00590
              WNTY  TEMM-1                                               03378 38 03087 00100
              WATY  TMM                                                  03390 39 03091 00100
              WATY  OUTPUT                                               03402 39 03473 00100
              S     AREA,N                                               03414 22 00446 00511
```

```
 OUT2   BC4  *+24                                                    03426 46 03450 00400
        BNR  OUT1,AREA,11                                           03438 45 03258 00440
        B    OUTX,,6                                                03450 49 03460 00000
 OUTX   DS   5                                                      03466 00005
 OUTC2  DS   5                                                      03471 00005
 OUTPUT DAS  11                                                     03473 00022
* OVERALL SHARP SUBROUTINE
*         SHARP THE LIST GIVEN BY RRPL FROM THE LIST GIVEN BY
*         TABL.   THE RESULT WILL BE SUBSUMED AFTER ONE TERM IS
*         SHARPED FROM ANOTHER
*         TABL WILL BE THE ADDRESS OF THE LAST CUBE OF THE RESULTS
*         IF THE AREA IS COMPLETELY SHARPED.  TABL WILL BE THE ADDRESS
*     THE RECORD MARK AT THE BEGINNING OF THE AREA
 SHARP  BV   *+12                                                   03494 46 03506 01400
        BNR  *+24,RRPL,11                                           03506 45 03530 0045J
        B    RX,,6                                                  03518 49 0374N 00000
        BNR  *+24,TABL,11                                           03530 45 03554 0046J
        B    RX,,6                                                  03542 49 0374N 00000
        TF   TERM,RRPL                                              03554 26 00456 00451
 SHR4   TF   TAB,TABL,,          TABL HAS ADDRESS OF LIST TO BE SHARPED FR   03566 26 00471 00461
 SHR3   TFM  SHPTX,SHR1,,        RRPL HAS ADDRESS OF LIST TO DO SHARPING W   03578 16 04134 -3602
        B    SHR                                                    03590 49 03746 00000
 SHR1   BNR  SHR5,TABE,11                                           03602 45 03626 00460
        B    RX,,6                                                  03614 49 0374N 00000
 SHR5   TF   TSSS,TABE                                              03626 26 00486 00466
        TF   TSS,TABL                                               03638 26 00491 00461
        TFM  SUMEX,SHR2,7                                           03650 16 04674 -3674
        B    SUME                                                   03662 49 04238 00000
 SHR2   TF   TABL,TSSS                                              03674 26 00461 00486
        S    TAB,N                                                  03686 22 00471 00511
        BNR  SHR3,TAB,11                                            03698 45 03578 0047J
        S    TERM,N                                                 03710 22 00456 00511
        BNR  SHR4,TERM,11                                           03722 45 03566 00450
        B    RX,,6                                                  03734 49 0374N 00000
        DORG *-4                                                    03741
 RX     DS   5                                                     03745 00005
* SINGLE SHARP SUBROUTINE
*         THE CUBE WHOSE ADDRESS IS IN TERM IS SHARPED FROM
*         THE CUBE WHOSE ADDRESS IS IN TAB
 SHR    TF   TABE,TABL                                             03746 26 00466 00461
        TR   TABLE,SHRPTB                                           03758 31 00300 04136
```

```
          TF   TEST,TAB,11                                                    03770 26 00526 0047J
          A    TEST,TERM,11,    TEST TO SEE IF SHARPING IS TO BE PERFORMED    03782 21 00526 00450
          TR   TABLE,ADDTB,,    IF THERE IS AN OVERFLOW FROM THIS ADD THERE   03794 31 00300 00130
          BNV  SHP3,,,         NO SHARPING TO BE DONE.                        03806 47 03830 01400
          B    SHPTX,,6                                                       03818 49 0413M 00000
SHP3      TF   HOLD,TAB,11,     THE CUBE OF TERM WILL SHARP THE CUBE OF TAB   03830 26 00548 0047J
          TF   SHP2+6,TAB,,     THE LIST.--THE NEW CUBES FROM THE SHARP WIL   03842 26 03896 00471
          S    SHP2+6,N1,,      BE PUT AT THE END OF THE LIST.                 03854 22 03896 00516
          TF   SHP2+11,TAB                                                    03866 26 03901 00471
          A    SHP2+11,ONE                                                    03878 21 03901 00590
SHP2      TR   0,0                                                            03890 31 00000 00000
          S    TABL,N                                                         03902 22 00461 00511
          TF   SOL+1,ZSOL,11                                                  03914 26 00538 0050J
SHP1      TR   TABLE,SHRPTB,,     SHARPING ACTION                           03926 31 00300 04136
          A    TEST,SOL,,       CLEAR OUT THE PREVIOUS BINARY VARIBLE        03938 21 00526 00537
          TDM  TEST+1,1,,       INSERT *ON** ONE POSITION TO RIGHT OF RESUL  03950 15 00527 00001
          A    SOL+1,TEST+1,,   ESTABLISH THE NEXT BINARY VARIBLE            03962 21 00538 00527
          BV   RET1,,,          TEST FOR LAST CUBE OF RESULT                 03974 46 04046 01400
          TF   TABE,HOLD,6,,    TRANSFER CUBE TO BE SHARPED TO NEW POSITION  03986 26 00460 00548
          A    TABE,SOL,6,      TRANSFORM TO NEW CUBE                        03998 21 00460 00537
          TR   TABLE,ADDTB                                                    04010 31 00300 00130
          A    TABE,N                                                         04022 21 00466 00511
          B    SHP1                                                           04034 49 03926 00000
RET1      TR   TABLE,ADDTB                                                    04046 31 00300 00130
          S    TABE,N1                                                        04058 22 00466 00516
          TD   TABE,RM,6,       PLACE RECORD MARK AT END OF TABLE            04070 25 00460 00401
          S    TABE,ONE                                                       04082 22 00466 00590
          B    SHPTX,,6                                                       04094 49 0413M 00000
RET       TR   TABLE,ADDTB                                                    04106 31 00300 00130
          B    SHPTX,,6                                                       04118 49 0413M 00000
SHPTX     DS   5                                                              04134 00005
*     THE SUBSTITUTE ADD TABLE FOR SHARPING ACTION
          DC   2,00                                                           04136 00002
          DC   11,02222070910                                                 04147 00011
          DC   2,-00                                                          04149 00002
          DC   2,00                                                           04151 00002
          DC   16,-0222222222222230                                           04167 00016
          DC   2,00                                                           04169 00002
          DC   2,00                                                           04171 00002
          DC   16,0222222222222258                                            04187 00016
          DC   2,06                                                           04189 00002
```

```
          DC   2,00                                                    04191 00002
          DC   15,012362222220222                                      04206 00015
          DC   20,02222222822222222202                                 04226 00020
          DC   10,022222222@                                           04236 00010
          DORG *-100                                                   04136
 SHRPTB   DSS  101                                                     04136 00101
*     SUBSUMING SUBROUTINE
*     SUBSUME ALL TERMS FROM TSSS TO TSS AGAINST ALL
*     TERMS IN THE TABLE FROM TSS TO BEGINNING OF TABLE
 SUME     TF   TASS,TSSS                                               04238 26 00476 00486
          TF   TAS,TSS                                                 04250 26 00481 00491
 SU5      C    TAS,TASS                                                04262 24 00481 00476
          BE   SUMEX,,6                                                04274 46 0467M 01200
          BH   SUMEX,,6                                                04286 46 0467M 01100
          BNR  SU2,TAS,11                                              04298 45 04322 0048J
          B    SUMEX,,6                                                04310 49 0467M 00000
 SU2      TR   TABLE,SUMTB                                             04322 31 00300 04676
          C    TAS,TASS,611,    DOES TAS CUBE COVER TASS CUBE           04334 24 0048J 00470
          BH   SU1                                                     04346 46 04454 01100
          C    TASS,TAS,611,    DOES TASS CUBE COVER TAS CUBE           04358 24 00470 0048J
          TR   TABLE,ADDTB                                             04370 31 00300 00130
          BH   SU10                                                    04382 46 04550 01100
          S    TAS,N                                                   04394 22 00481 00511
          BNR  SU2,TAS,11                                              04406 45 04322 0048J
 SU6      TF   TAS,TSS                                                 04418 26 00481 00491
          S    TASS,N                                                  04430 22 00476 00511
          B    SU5                                                     04442 49 04262 00000
 SU1      TR   TABLE,ADDTB,,    TAS CUBE DOES COVER TASS CUBE          04454 31 00300 00130
          TF   SU4+6,TASS,,     REMOVE TASS CUBE FROM LIST             04466 26 04520 00476
          S    SU4+6,N1                                                04478 22 04520 00516
          TF   SU4+11,TASS                                             04490 26 04525 00476
          A    SU4+11,ONE                                              04502 21 04525 00590
 SU4      TR   0,0                                                     04514 31 00000 00000
          S    TSSS,N                                                  04526 22 00486 00511
          B    SU6                                                     04538 49 04418 00000
 SU10     TF   SU11+6,TAS,,     TASS CUBE DOES COVER TAS CUBE          04550 26 04604 00481
          S    SU11+6,N1,,      REMOVE TAS CUBE FROM LIST              04562 22 04604 00516
          TF   SU11+11,TAS                                             04574 26 04609 00481
          A    SU11+11,ONE                                             04586 21 04609 00590
 SU11     TR   0,0                                                     04598 31 00000 00000
          S    TSSS,N                                                  04610 22 00486 00511
```

```
          S     TASS,N                                                  04622 22 00476 00511
          S     TSS,N                                                   04634 22 00491 00511
          S     TAS,N                                                   04646 22 00481 00511
          B     SU5                                                     04658 49 04262 00000
 SUMEX    DS    5                                                       04674 00005
*      SUBSTITUTE ADD TABLE FOR SUBSUMING OPERATION
          DC    20,2222222222222222222                                  04694 00020
          DC    18,222222222222222222                                   04712 00018
          DC    18,222222222222222222                                   04730 00018
          DC    2,22                                                    04732 00002
          DC    2,22                                                    04734 00002
          DC    2,2@                                                    04736 00002
          DORG  *-60                                                    04676
 SUMTB    DSS   61                                                      04676 00061
 RM       DC    1,@,401                                                 00401 00001
 NOTE1    DAC   34,START OF PROGRAM GIVE CUBE LENGTH@                   04739 00068
 NOTE2    DAC   17,LIST OF COCYCLES@                                    04807 00034
 NOTE3    DAC   33,ILLEGAL INPUT - PLEASE TRY AGAIN@                    04841 00066
 NOTE4    DAC   15,END OF PROBLEM@                                      04907 00030
 NOTE5    DAC   5,VOID@                                                 04937 00010
 TNTN1    DAC   17,LIST OF ON TERMS@                                    04947 00034
 TNTN2    DAC   24,LIST OF DONT CARE TERMS@                             04981 00048
 TNTN3    DAC   31,A TWO LEVEL MINIMIZED SOLUTION@                      05029 00062
 TNTN4    DAC   18,LIST OF OFF TERMS@                                   05091 00036
          DC    1,@                                                     05126 00001
 FREE     DS    1                                                       05127 00001
*      REGULAR ADD TABLES
*      NEEDED TO RESTORE ADD TABLES AFTER LOGIC OPERATIONS
          DC    20,00123456789123456789                                 05147 00020
          DC    10,-0234567890                                          05157 00010
          DC    8,13456789                                              05165 00008
          DC    2,-01                                                   05167 00002
          DC    8,-24567890                                             05175 00008
          DC    2,-12                                                   05177 00002
          DC    6,356789                                                05183 00006
          DC    2,-01                                                   05185 00002
          DC    2,-23                                                   05187 00002
          DC    6,-467890                                               05193 00006
          DC    2,-12                                                   05195 00002
          DC    2,-34                                                   05197 00002
          DC    4,5789                                                  05201 00004
```

```
            DC    2,-01                                    05203 00002
            DC    2,-23                                    05205 00002
            DC    2,-45                                    05207 00002
            DC    4,-6890                                  05211 00004
            DC    2,-12                                    05213 00002
            DC    2,-34                                    05215 00002
            DC    2,-56                                    05217 00002
            DC    2,79                                     05219 00002
            DC    2,-01                                    05221 00002
            DC    2,-23                                    05223 00002
            DC    2,-45                                    05225 00002
            DC    2,-67                                    05227 00002
            DC    2,8@                                     05229 00002
            DORG  *-100                                    05129
ADDT        DSS   101                                      05129 00101
ADDTB       DS    100,130                                  00130 00100
TABLE       DSS   100,300,          ADD TABLES             00300 00100
BEGIN       TR    ADDTB,ADDT                               05230 31 00130 05129
            RCTY                                           05242 34 00000 00102
            BC4   TLM                                      05254 46 00600 00400
            WATY  HEAD1                                    05266 39 05699 00100
            RCTY                                           05278 34 00000 00102
            BC4   TLM                                      05290 46 00600 00400
            WATY  HEAD2                                    05302 39 05783 00100
            RCTY                                           05314 34 00000 00102
            BC4   TLM                                      05326 46 00600 00400
            WATY  HEAD3                                    05338 39 05827 00100
            RCTY                                           05350 34 00000 00102
            BC4   TLM                                      05362 46 00600 00400
            WATY  HEAD4                                    05374 39 05925 00100
            RCTY                                           05386 34 00000 00102
            BC4   TLM                                      05398 46 00600 00400
            WATY  HEAD5                                    05410 39 06021 00100
            RCTY                                           05422 34 00000 00102
            BC4   TLM                                      05434 46 00600 00400
            WATY  HEAD6                                    05446 39 06137 00100
            RCTY                                           05458 34 00000 00102
            BC4   TLM                                      05470 46 00600 00400
            WATY  HEAD7                                    05482 39 06257 00100
            RCTY                                           05494 34 00000 00102
            BC4   TLM                                      05506 46 00600 00400
```

```
            WATY HEAD8                                                  05518 39 06385 00100
            RCTY                                                        05530 34 00000 00102
            BC4  TLM                                                    05542 46 00600 00400
            WATY HEAD9                                                  05554 39 06455 00100
            RCTY                                                        05566 34 00000 00102
            BC4  TLM                                                    05578 46 00600 00400
            WATY HED10                                                  05590 39 06573 00100
            RCTY                                                        05602 34 00000 00102
            BC4  TLM                                                    05614 46 00600 00400
            WATY HED11                                                  05626 39 06653 00100
            RCTY                                                        05638 34 00000 00102
            BC4  TLM                                                    05650 46 00600 00400
            WATY HED12                                                  05662 39 06719 00100
            RCTY                                                        05674 34 00000 00102
            B    TLM                                                    05686 49 00600 00000
HEAD1       DAC  42,THIS IS A TWO LEVEL MINIMIZATION PROGRAM.@          05699 00084
HEAD2       DAC  22,  INPUTS REQUIRED ARE@                              05783 00044
HEAD3       DAC  49,1 SIZE OF CUBES. FIVE DIGITS WITH LEADING ZEROS.@   05827 00098
HEAD4       DAC  48,2 CUBICAL ARRAYS. 1 FOR THE ON CONDITION, 0 FOR@    05925 00096
HEAD5       DAC  50,  THE OFF CONDITION, AND - FOR THE UNSPECIFIED CON  06021 00100
            DAC  8,DITION.@                                             06121 00016
HEAD6       DAC  49,3 FIRST METHOD, GIVE THE ON CUBICAL ARRAY AND THE   06137 00098
            DAC  11, DONT CARE@                                         06235 00022
HEAD7       DAC  50,  CUBICAL ARRAY, OR BY USING SENSE SWITCH 1 GIVE T  06257 00100
            DAC  14,HE ON CUBICAL@                                      06357 00028
HEAD8       DAC  35,  ARRAY AND THE OFF CUBICAL ARRAY.@                 06385 00070
HEAD9       DAC  50,4 AFTER THE LAST CUBE HAS BEEN ENTERED IN A CUBICA  06455 00100
            DAC  9,L ARRAY,@                                            06555 00018
HED10       DAC  40,  ENTER A RECORD MARK AS THE LAST CUBE.@            06573 00080
HED11       DAC  33,5 ENTER ONLY ONE CUBE PER INPUT.@                   06653 00066
HED12       DAC  49,6 USE SENSE SWITCH 4 TO SKIP ANY CUBE THAT IS NOT   06719 00098
            DAC  9, WANTED.@                                            06817 00018
            DEND BEGIN                                                  05230

06834 CORE POSITIONS REQUIRED
00525 STATEMENTS PROCESSED
```

VITA

Robert Whitney Butler

Candidate for the Degree of

Master of Science

Thesis: TWO LEVEL BOOLEAN MINIMIZATION USING A SMALL DIGITAL COMPUTER

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Fort Worth, Texas, January 18, 1936, the son of Corwin A. and Edith P. Butler.

Education: Graduated from Blackwell Senior High School, Blackwell, Oklahoma in May 1954; Received the Bachelor of Science degree from Oklahoma State University, with a major in Electrical Engineering in January, 1959; Completed requirements for the Master of Science degree, with a major in Electrical Engineering in May 1966.

Professional experience: Employed by International Business Machines Corporation in February 1960 as a programmer; Entered ine United States Army in March 1960; Returned to International Business Machines in June 1962.